rg-azure-analytics rg-azure-analytics

* Setup Fabric Capacity in Power BI workspace

A screenshot of a computer

Description automatically generated

**Project Overview**

This project aims to develop a news dashboard that displays the latest news articles and performs sentiment analysis on them. Data will be ingested from Bing Search using the Bing Search API.

**Components Involved**

* **Microsoft Fabric:** This is the primary tool used for the entire project. It includes various components that handle data ingestion, transformation, analysis, and visualization.
* **Bing Search API:** This API is used to retrieve news data from Bing Search.
* **Data Factory:** This tool within Fabric helps connect to the Bing Search API and ingest news data in JSON format into the Azure Data Lake Storage (ADLS).
* **Azure Data Lake Storage (ADLS):** This acts as the storage solution for the project. It stores the raw JSON data ingested from Bing Search.
* **Synapse Data Engineering:** This tool is used to process the raw JSON data into a structured Delta table format.
* **Synapse Data Science:** This component performs sentiment analysis on the processed Delta table using a pre-trained machine learning model.
* **Power BI:** This tool is used to create reports and visualizations based on the sentiment-analyzed data.

**Steps Involved**

1. **Data Ingestion:**
   * Use Data Factory to connect to the Bing Search API.
   * Configure the Bing Search API to use the search term "latest news" (this can be customized later).
   * Injest the retrieved news data in JSON format into the ADLS.
2. **Data Transformation:**
   * Use Synapse Data Engineering to process the raw JSON data.
   * Transform the JSON data into a clean and structured Delta table format.
   * Store the Delta table back in the ADLS.
3. **Data Analysis:**
   * Use Synapse Data Science to read the processed Delta table.
   * Perform sentiment analysis on the news articles using a pre-trained machine learning model based on the description column.
   * Store the sentiment-analyzed data as a new Delta table in the ADLS.
4. **Data Visualization:**
   * Use Power BI to create reports and visualizations based on the sentiment-analyzed data.
   * Configure the Power BI report to display only the latest news articles published in the last 24 hours.
5. **Scheduling and Alerts:**
   * Create a pipeline in Data Factory to orchestrate all the tasks involved in data ingestion, transformation, analysis, and visualization.
   * Schedule the pipeline to run daily at 6:00 AM using Data Factory.
   * Configure alerts in Data Activator to notify you when specific conditions in the data are met (e.g., positive sentiment percentage greater than zero).

**Conclusion**

By following these steps, you will be able to build a complete end-to-end Azure data engineering project using Microsoft Fabric. This project will help you gain practical experience with various Azure data engineering tools and services.

**Additional Notes**

* The instructor recommends implementing the project yourself to gain a deeper understanding of the concepts.
* The video is part of a series on Azure data engineering. The instructor encourages viewers to like and subscribe to the Mr. K Talks Tech channel for more content.

Architecture:

* We will be using Lake Database for storing the raw JSON file.
* Data factory will be used for to create pipelines for orchestration.
* Once stored, we will transform it to proper table structure with a predefined schema. This will be done by Synapse Data Engineering and will be stored in **delta** table.
  + Stage 1 : raw data in files
  + Stage 2 : Tabular data in delta table
* With Synapse Data Engineering, we will be creating spark notebooks for transformation
* Once the data in cleaned and the sentiment analysis has been done, data associated with the sentiment will be stored as a delta table as well.
* For sentiment analysis, we will use Synapse Data Science.
* Delta tables with the sentiment predictions will be also stored in the data lake.

A diagram of a company

Description automatically generated

Agenda

A black background with white text

Description automatically generated

**START**

* 1. **Environment Setup**
* Create a resource group, create a tag for it.
* Create a bing API with minimum pricing and put it under the same tag
* Go to Power Bi workspaces and create a new workspace for your project. In setting select “trial” license or whichever you prefer.
* Go to Power Bi workspace where we have created the Fabric workspace and switch to “Data Engineering” from the Bottom left.
* Select “lakeHouse” and create a new database (bing\_lake\_db in our case)
  1. **Data Ingestion**
* Move to Microsoft fabric and then to the workspace we created.

A screenshot of a computer

Description automatically generated

* From bottom left men select **data Factory**
* Select “copy data>> add to canvas” activity and then configure it.
  + Under source >> More >> “Search rest and select”>>
  + For the connection end point use the bing news api link. You can get this from the link “https://learn.microsoft.com/en-us/bing/search-apis/bing-news-search/reference/endpoints >> test connection and apply
  + We need to give this fabric necessary authentication to access the data from the Bing API (got to Header section in bing API documentation)
  + Copy the **“required Header” and** then head to the advanced section of Source setting
  + Go to “**Additional Header” >>** “New Header” and then paste the required header in the **“Name”**
  + Move to the bing API page >> **keys and endpoints >> copy key1 >> paste in** value on the source configuration page
  + Move to Relative Url column and then we need to configure it for getting the **“latest news”** then paste **“?q=latest+news”**  in the parameter
  + Check preview.
  + Different query parameters can be added. Find these in the query parameters section of the bing API documentation and then modify the relative URL to

**“?q=latest+news&count=100&freshness=Day&mkt=en-US”**

* Move to destination section and select the “bing\_lake\_db” created.
* We will be copying all the raw data to the lake house in json file format
* Configure as below

A screenshot of a computer

Description automatically generated

* Leaving the “Directory” column blank will put the data at the top level.
* Save and “Run”.
* Open the bing\_lake\_db and refresh.
* The file section should have the newly ingested data
  1. **Data Transformation (Incremental Load)**
* We will use synapse data engineering for this step.
* Make sure are in Data Engineering tab from the bottom left. Go to Home from top left and create a notebook.
* Rename the notebook and attach the Lakehouse database to it.
* Transformation will be done with Spark. Initiate the below for reading the file

**A screenshot of a computer

Description automatically generated**

* **“df” :** temporary table that looks like a table structure with proper rows and columns.
* Microsoft fabric is a serverless compute system. This will process the data faster.
* Now we have all the Json objects in a single row and we need to transform this data frame such that each article exists in a single row. This would be done using the explode function in pyspark.

**https://spark.apache.org/docs/3.1.3/api/python/reference/api/pyspark.sql.functions.explode.html**

* once run, we can have separate rows for all the news articles. Since the definition has been set to 100. We can upto 100 articles.
* Steps are followed for transformation to extract data from json strings.
* json\_list = df\_exploded.toJSON().collect() converts a Pandas DataFrame df\_exploded into a list of JSON strings and stores it in the variable json\_list.
* Json string data will be parsed to valid python dictionary using json.loads()
* Exception encountered during processing of the JSON data:

**A screenshot of a computer program

Description automatically generated**

* If conditions will be used to exclude these cases and get a more consistent data set
* Next we format the date

**A screenshot of a computer

Description automatically generated**

* The transformed data is written to LakeHouse DB in delta Format
  + df\_cleaned\_final.write.format("delta").saveAsTable("bing\_lake\_db.tbl\_latest\_news")
* Now this data can be queried from a table. With this we completed our processing of raw JSON file to a clean and structure table using Spark notebook and data transformation steps.
* We will move forward to incremental load now.
* Utilizing SQL Merge Data Warehousing. We will configure the incremental load for SCD1. This is because this project is focused towards creating reports on latest news and hence keeping the previous versions is not greatly productive.

**A screenshot of a computer program

Description automatically generated**

* Exception Handling done in the above code since every time we rerun the code to populate fresh data to the table, the table might pre-exist and hence will fail to create. Since we are working with SCD1 and we need new data always, in order to deal with analysis exception, we will simply recreate the table every time.

Same handled via Exception Handling in the above code.

* **Table Populated**

**A screenshot of a computer

Description automatically generated**

* 1. **Sentiment Analysis (Incremental Load)**
* Move to Data Science tab from the bottom left menu on the Fabric Work Space
* We will be using a pre-trained Machine Learning Model. So far, we have built the pipeline, fetched the data using API and transformed it. Now, we will be working on the data scientist part, since the data engineering part is complete now.
* Create a new notebook from the Data Science Page and the add the lake House DB we have created
* Load the data with Spark.
* We will be using 1 of the pre-built intelligent model from **SYNAPSE ML (previously known as MMLSpark).**
* We will be using the description column to predict the sentiment.
* **Model Configured**

A screenshot of a computer

Description automatically generated

* Model creates two new columns. The response column hold the json output of the model. The output has the sentiment.

A screenshot of a computer

Description automatically generated

* Sentiment value extracted and “error” and “response” column dropped from the final data frame.

A screenshot of a computer

Description automatically generated

* Now, we will insert this into bing\_lake\_db. This wil create a new table. This will be further used for report generation.

**A screenshot of a computer

Description automatically generated**

* With this the sentiment analysis part is completed. Now we will head on over to the reporting part of this project. We will be using Power BI for this task.
  1. **Data Reporting**
* We will start with building the semantic model for our table to be used for report by Power BI. **“tbl\_sentiment\_analysis”**
* Head to the Gen 2 Lake, **bing\_lake\_db** . Click on **“New Semantic Model”.** 
  + Select **“tbl\_sentiment\_analysis”.**

A screenshot of a computer

Description automatically generated

* Go to the data model, change the category to “web url”

A screenshot of a computer

Description automatically generated

* Head to Power BI and create report as below. Create filter as below so that only the latest news is displayed in the report pane. We can still get the report from other dates via the slicer.

A screenshot of a computer

Description automatically generated

* Move to data model. We need to create some measures here to know the daily overall sentiment of the news.
* Head to semantic model and create 3 measures as below

A screenshot of a computer

Description automatically generated

* Add cards on the POWER BI report for the sentiments and configure them also to display sentiment for only the latest report.

A screenshot of a computer

Description automatically generated

* Report Completed. Lets move to building some pipelines
  1. **Building Pipelines**
* Head to the “news-ingestion-pipeline” pipeline we built earlier to fetch the latest news into the lake house database in raw Jason file.
* Since we already have a notebook for data transformation, we will be using it to automate our process.
* Select a “Notebook” activity from the top menu and make connection to previous such that it runs once the previous activity has successfully completed.
* Configure the pipeline as below
* The first activity will transform the pipeline and the second activity will perform sentiment analysis and load it to the lake house database.

A screenshot of a computer

Description automatically generated

* We will parametrize the source activity. The “relative url” setting for the source mentions “latest news”. Instead of this we will create a parameter to specify the topic of the news. The remaining part of the “relative url” will ensure that we get the latest data.

A screenshot of a computer

Description automatically generated

* Select “Source” activity. Click add dynamic content and then modify the url as below “?q=@{pipeline().parameters.search\_term}&count=100&freshness=Day&mkt=en-US”
* Test it and then schedule the pipeline

A screenshot of a computer

Description automatically generated

* The tbl\_sentiment\_analysis table has been modified for schema changes with the notebook. We need to reflect the same on the semantic model. Head on to the data model and select **“edit table”**.
* Select refresh and submit.
* Now in PowerBI report, since we have used **“datePublished”** for the filter and it is date type column after making all the changes. We will get date based filter options.
* These options were not present earlier as the column was stored just a text column. With any categorization.

**A screenshot of a computer

Description automatically generated**

* Reporting Completed.
  1. **Setting up alerts (Data Activator)**
* Switch to data activator Tab from the bottom left menu
* Click on “Reflex(Preview)” . We would be using to manage and monitoring all the alerts we will be creating using the Data Activator.
* Configure Data Activator as below. You get to set the alert from the options for each graph item.

A screenshot of a computer

Description automatically generated

* Alerts look as below for in the Data Activator window.

A screenshot of a computer

Description automatically generated

* 1. **End to End Testing**
* Move to the Data Engineering section from the bottom left menu. We will do the complete end-to-end pipeline testing.
* Select **“new-ingestion-pipeline”**
* Click run and provide option “movies” to the parameter.
* Power BI report updates automatically according to the new data.
* Testing Concluded successfully.

**PowerBI Reoport Page 1**

**A screenshot of a computer

Description automatically generated**

**PowerBI Report Page 2**

**A screenshot of a computer

Description automatically generated**